Global Motion Estimation from Pan-Tilt Cameras

Roman Bachmann*
Supervised by: Helge Rhodin,† Pascal Fua‡

Computer Vision Lab
EPFL
Lausanne / Switzerland

Abstract

Coaches in alpine skiing would like to know the speed and useful biomechanical variables at each turn in a run. Existing methods using body-worn sensors are distracting and marker-based manual image annotation for inference is time consuming. We propose a method of estimating an athlete's global 3D pose using multiple cameras. First, tight estimated bounding boxes of the skier are fed to a 2D pose estimator network. The 3D pose is then reconstructed using a bundle adjustment method. We show results both when using fully calibrated cameras, as well as when estimating the rotation of Pan-Tilt-Zoom cameras. To overcome shortcomings of existing datasets we created a new alpine skiing dataset and trained all methods on it. Our method estimates accurate global 3D poses from images only, providing coaches with an automatic and fast tool to improve an athlete's performance.

Keywords: Computer Vision, Pose Estimation, Motion Capture, Camera Calibration, Deep Learning

1 Introduction

In many winter sports like alpine skiing, coaches would like to know performance metrics such as Center of Mass, speed and various biomechanical variables at every point in time, giving them accurate feedback about potential increases or losses in speed. This can then be used to enhance the athlete's performance at every turn in the run. Existing methods like optical barriers only offer average speeds within segments, while other methods using Inertial Measurement Unit (IMU) sensors are cumbersome for the athletes to wear. Using motion capture suits is also not feasible in high-speed settings with wide baselines.

Existing methods using only one camera are only able to reconstruct a 3D pose relative to the camera's coordinate frame, which contains no information about the athletes global position and speed. If the camera rotates, so does it's coordinate system. We therefore set ourselves the goal of estimating an athlete's global 3D pose at every point in time using just video frames from multiple cameras arranged around the track. One way to get those poses is to manually annotate every frame and recreate the 3D structure from that. This manual annotation is however very tedious and time-consuming, so instead we chose to train a pose estimation network to predict 2D joint locations without the athletes needing to wear markers. Normally, pose estimation algorithms are only trained on human pose databases which don't contain skiing images, and if they do, the skis and poles are not annotated. Existing alpine skiing datasets [16, 7, 15] are very limited in the number of athletes and locations that they feature, making methods trained on them not generalize well. To remedy those problems we created a new alpine skiing dataset, containing 1982 manually annotated frames showing 32 athletes in diverse weather conditions.

To go from videos to 3D poses, we propose the following method: Because skiers are often very small in the captured images, we first train a network to predict a tight bounding box around them. Those crops are then given to the pose estimation network that was trained from scratch on the new skiing dataset. The 2D detections from all cameras are then combined in a bundle adjustment approach to reconstruct the global 3D pose. We compare the performance of taking fully calibrated cameras with a method estimating also the camera rotations, i.e. the direction the cameras are facing.

2 Related work

This paper builds upon work in the fields of object detection, as well as 2D and 3D pose estimation techniques. In the following, we outline the most important advances in each of those fields.

2.1 Object detection

Object detection is the process of localizing occurrences of certain classes in images and outputting a tight bounding box around them. In recent years, several Deep Learning approaches made great advances in terms of accuracy and detection speed.

Redmon *et al.* introduced the YOLO [14] algorithm that instead of applying a neural network to multiple scales of

*roman.bachmann@epfl.ch
†helge.rhodin@epfl.ch
‡pascal.fua@epfl.ch

an image runs it through it only once, making it very fast at test time.

Liu *et al.* propose the Single Shot MultiBox Detector (SSD) [9] generating scores for the presence of an object in predefined bins and then adjusting the bins to better match the object shape. Detections from multiple feature maps and different resolutions are then combined to allow for detection of various sizes in one single network stage.

2.2 2D human pose estimation

To get an understanding of the human pose in images, Deep Learning based pose estimation algorithms are tasked to find the 2D locations of all specified body joints. Cao *et al.* take a multi-stage approach with OpenPose [2], predicting confidence map for each joint and a Part Affinity Field (PAF) for each limb (a vector field encoding the association of two joints connected by a limb). Using the Part Affinity Fields as an indicator for which joints in the heatmaps belong together, the poses of multiple people can be efficiently differentiated.

2.3 Global 3D human pose estimation

Using at least two cameras from different perspectives, it is possible to obtain a global 3D pose estimate and potential ambiguities in scale can be resolved.

Several papers [3, 4, 13] leverage common line markings of sports fields as known reference points for pan-tilt-zoom (PTZ) camera calibration. Those methods can also leverage the geometric constraints that games like football are played on a two-dimensional surface with a limited spatial extent.

Pavlakos *et al.* [11] propose to extend pictorial structure models by taking CNN generated 2D heatmaps and resolving the 3D structure in a quantized grid by maximizing a likelihood term explaining the 2D detections. They use however known camera parameters.

Puwein *et al.* [12] jointly estimate a 3D human pose and the position and orientation of several fixed wide-baseline cameras using a bundle adjustment method that minimizes an energy function comprising reprojection errors, a smoothness term and optical flow consistency between the motion of the estimated kinematic structure and the videos.

Similarly, Elhayek *et al.* [6] estimate both pose and camera locations simultaneously, with the difference that some cameras are fixed, while others can freely move. They minimize an energy function containing a negative likelihood term describing the similarity of the model parameters to the measured data, as well as smoothness terms for both the human pose and the cameras.

Using multiple unsynchronized and uncalibrated cameras, Takahashi *et al.* [17] propose a bundle adjustment method that leverages the limb lengths as priors on the human body and takes into account that the 2D pose estimations contain some amount of error. To this end, they relax the reprojection error, penalizing errors up to a certain point less.

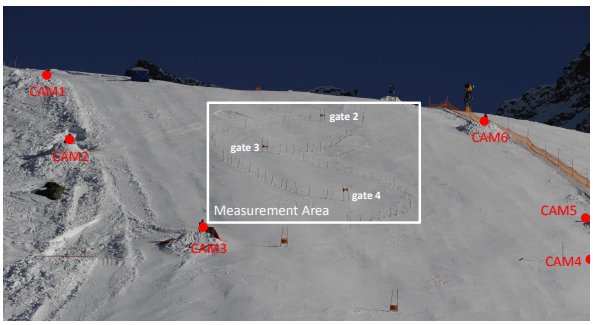


Figure 1: Setup overview of the multi-view skiing dataset.

3 Datasets overview

While there exists extensive datasets for human poses in various settings like the MPII Human Pose [1] dataset or the Human3.6M [8] dataset, they feature only none or very few skiing images and lack annotation of the skis and poles. Additionally they feature even fewer images of professional athletes in a racing scenario, which would make inference more difficult in those cases. To train a 2D detector, we have therefore decided to create a new alpine skiing dataset featuring mainly semi-professional athletes. For the purpose of evaluating 3D pose estimation methods, we used a manually annotated multi-view pan-tilt-zoom alpine skiing dataset [16, 7, 15], which is explained in detail in the following section.

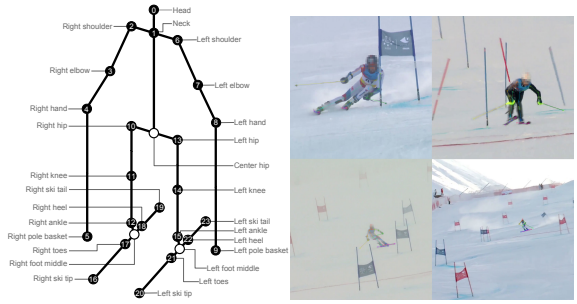
3.1 Kühtai multi-view alpine skiing dataset

The aforementioned multi-view alpine skiing dataset [16, 7, 15] features 6 professional athletes on a Giant Slalom slope with three turns, filmed by six cameras that are arranged in a circle around the center of the track as shown in Figure 1. 2D joint locations were manually annotated. Calibration points around the track served to calculate the camera parameters, specifically the intrinsic and extrinsic camera matrices. From this, global ground truth 3D poses were computed.

While the existing skiing dataset is well suited for developing semi-supervised models [15], the fact that it only features 6 different athletes in similar suits performing the run on the same slope with the same camera angles would make methods trained on it unable to generalize to different skiing settings.

3.2 New alpine skiing dataset

To improve upon generalization, we create a new large dataset for alpine skiing. We downloaded 16 alpine skiing videos that were posted on Youtube under the Creative



(a) Skier joints (b) Example images

Figure 2: Figure a shows the 24 annotated joints. The white circles are helper joints in the tree structure, with the center hip being the root joint. Figure b shows four annotated images of the new alpine skiing dataset.

Commons license, featuring mainly semi-professional ski racers from many different perspectives. Those videos were split into 147 training and 11 validation sequences, each being one continuous camera motion without any cuts. From each split, frames were sampled in fixed intervals ranging from 0.3 to 10 seconds, depending on the discipline. Slalom, featuring more variation in poses, had a higher sampling frequency than downhill where athletes often stay in the same pose for long stretches. In total, 1982 images were sampled and annotated with 24 key points (see Figure 2a), of which 1830 were used as training and 152 as validation images. The dataset comprises at least 32 unique athletes, 17 of which are women and 15 are men. It features 5 unique locations in various weather conditions ranging from sunny to foggy (see Figure 2b). There are 32 Slalom, 52 Giant Slalom, 26 Super-G, 24 Downhill and 6 training sequences filmed from a follow cam including scenes from very close to very far away. The dataset is available on request to the authors.

Calibration pole augmentation As this newly created dataset does not feature any calibration poles like the Kühtai dataset, evaluating a 2D pose estimation algorithm that was only trained on this will produce significant outliers (mainly with the ski poles). One way to improve robustness on the Kühtai dataset is to augment training images with randomly superimposed cutouts of eight different calibration poles (see Figure 3). At training time, we uniformly sample $\mathscr{U}(0,20)$ randomly selected poles and place them uniformly over the image. The poles are scaled by $\mathscr{U}(0.5,2.5)$ and rotated $\mathscr{U}(-15,15)$ degrees. We compare this method to adding one Kühtai sequence to the training of OpenPose.

4 Methodology

Our goal is to take as input a set of synchronized video streams of the same athlete filmed from different angles and estimate the global 3D pose. The cameras are assumed

Figure 3: Left: Example image showcasing calibration poles. Right: Augmented alpine skiing image.

to be calibrated, meaning we know the intrinsic and extrinsic matrices for each frame and camera. Experiments were also done inferring the rotation matrix of each camera. To go from images to 3D pose, we propose a multi-staged pipeline as shown in Figure 4, where 2D pose detections are generated from cropped images around the athlete and then 3D poses are optimized to best fit all localized 2D joints. Generating 2D estimations first allows us to analyze potential detection weaknesses when using a new dataset, before developing a method for 3D joint detection.

First, we run an object detection network on each video stream to generate a tight square bounding box around the skier. Outliers are filtered out and bounding box detections are temporally smoothed. The OpenPose 2D pose estimation network is run on the square crop, generating joint heatmaps and Part-Affinity Fields, from which 2D joint key points are extracted. The 2D detections from all cameras are then incorporated into a bundle adjustment method which reconstructs the underlying 3D pose of the skier.

4.1 Skier detection

Sports videos with wide baselines often contain sequences of very different zoom levels, resulting in the athlete size in the image ranging from frame-filling to only making up a very small portion of the frame. While pose estimation networks like OpenPose run images through their network at multiple scales to account for this fact, athletes can be so small in images that detection fails completely. Even when the skier fills a good portion of the frame and the input scale is right for the pose estimation network, other people on the slopes or other high-contrast objects in the background can lead to wrong detections when only interested in the pose of the main subject. We therefore first detect a tight bounding box around the athlete, resulting in the pose estimation network always receiving examples of the same scale.

For this task we chose the Single Shot MultiBox Object Detector (SSD) [9] network (described in Section 2.1) for its good performance and low computational overhead during both train and test time. As the SSD network is a

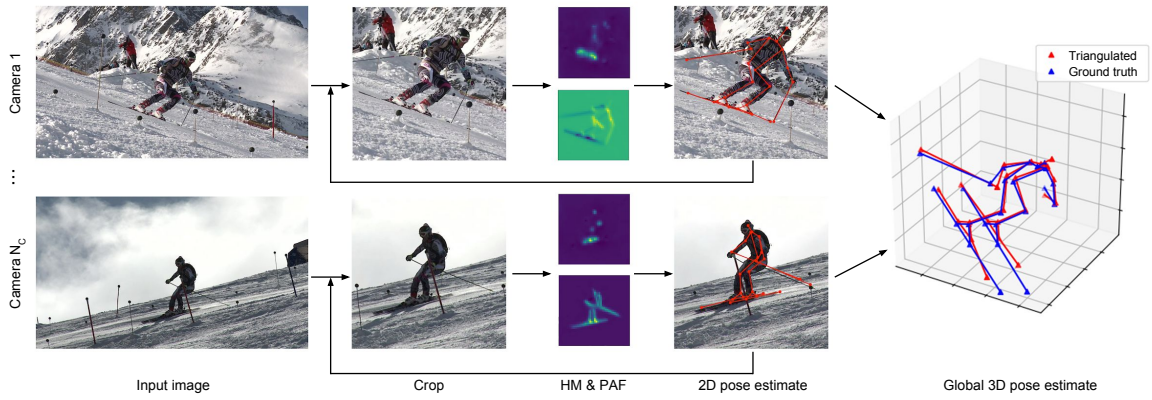


Figure 4: Method pipeline overview: Images from different cameras are preprocessed to find 2D joint locations by first cropping a bounding box around the athlete and then running a pose estimation network on it. Pose estimates are used to refine the bounding box. From all N_c cameras frames and over all time steps, the global 3D position is found using a bundle adjustment method.

single-image detector and does not incorporate temporal information from the fact that we are dealing with videos, detections between frames can suffer from jitter and outliers if other people are present in the scene. To remove strong outliers, we replace all detections whose center deviates beyond the bounding box of the last correctly detected frame by the latter. Jitter and varying sizes of the bounding boxes are dealt with by applying strong Gaussian smoothing to the center location of the crops and their respective side lengths with parameters $\sigma_{center} = 10$ and $\sigma_{side} = 50$. Finally, side lengths are scaled by a factor of 1.5 for the newly created and 2 for the Kühtai dataset and all crops get resized to 500 x 500 pixels.

We trained a PyTorch implementation of the SSD network [5] on the newly annotated alpine skiing dataset for 2000 iterations, using a batch size of 32 images. The network was initialized with pretrained VGG 16 weights. The optimizer used was Stochastic Gradient Descent (SGD) with learning rate 0.001, momentum 0.9 and weight decay 0.0005. The learning rate was scaled by 0.1 at iterations 1000 and 1500. As this PyTorch implementation was made for multiple object classes we trained it with one athlete class and one unused dummy class.

4.2 2D pose estimation

Given an input image $I \in \mathbb{R}^{w \times h \times 3}$ of width w and height h, the task of 2D pose estimation is to compute x and y coordinates for every joint $j \in \{1,...,N_J\}$. The OpenPose network returns for every joint j a confidence/heat map $H_j \in \mathbb{R}^{w \times h}$ and for each limb/bone $l \in \{1,...,N_L\}$ a Part Affinity Field (PAF) $B_l \in \mathbb{R}^{w \times h \times 2}$, with every point in B_l encoding a vector describing limb orientations. Because in our problem we are focusing solely on the pose estimation of a single athlete on the slope and not multiple people, we don't rely on some of the multi-person detection advantages that PAF's bring to the table. Indeed, for this task we take the maximum location $p_j^* \in \mathbb{R}^2$ as $p_j^* = \arg\max_{w,h} H_j$ of each confidence map for each joint j.

Implementation Details For training OpenPose, all images were resized to 736 x 368 pixels. In addition to the augmentation with Kühtai calibration poles (as explained in Section 3.2), for data augmentation each of the following transformations was applied independently with a probability of 0.5:

- Adjusting the image gamma value uniformly by a factor of $\mathscr{U}(0.5, 1.5)$.
- Shifting the image hue uniformly by $\mathscr{U}(-15°, 15°)$.
- Rotating the image uniformly by $\mathscr{U}(-40°, 40°)$.
- Mirroring the image horizontally.

In addition, all training images were randomly cropped around the ground truth pose of the skier, such that the network always received poses of roughly the same scale. During test time, no data augmentation is applied besides resizing the imaged to the input resolution.

Training was done in batch sizes of 8 over 200 epochs using the Adam optimizer with learning rate 0.00004, momentum 0.9 and weight decay 0.0005. In all cases, the first OpenPose stage block was initialized with pretrained VGG 19 weights, while all other stages were either randomly initialized or using weights pretrained on the MPII dataset.

During test time the OpenPose model that yielded the lowest validation error during all training epochs was chosen. The outputs from the SSD network were resized to 368 x 368 pixels and run through OpenPose six times scaled by 0.5, 0.75, 1, 1.25, 1.5 and 2 for being able to detect a wider range of SSD outputs. Results from all scales were averaged to generate the final heatmaps and PAFs.

Refining SSD bounding boxes using OpenPose Because of the smoothing of detected SSD bounding boxes, drifts in the crop with respect of the athlete's center may still be present. To remedy this, we run OpenPose on the generated crops and take the median of the computed joint positions as the new center for each frame. Then, we apply a weaker Gaussian smoothing pass to the center locations and side lengths with parameters $\sigma_{center} = 5$ and $\sigma_{side} = 5$.

4.3 3D pose estimation

The last step in the pipeline is estimating the 3D poses of the skier using a bundle adjustment optimization method with the detected 2D joint locations. We take the OpenPose output $\boldsymbol{p}_j^{f,c} \in \mathbb{R}^2$ from all cameras $c \in \{1,...,N_C\}$, over all frames $f \in \{1,...,N_F\}$, for each joint $j \in \{1,...,N_J\}$ and reconstruct the respective underlying 3D joint positions $\boldsymbol{P}_j^f \in \mathbb{R}^3$ in global space. Let us denote the complete 3D pose at time f as $\boldsymbol{P}^f \in \mathbb{R}^{N_J \times 3}$.

We used two different ways of parameterizing the 3D pose: In the first method, every 3D joint position is simply described by its global 3D x, y, z location without any restrictions. The second possibility is modeling the athlete's pose as an articulated tree-like structure, parametrized by joint angles and using one global x, y, z coordinate to position the tree root. We used the tree model described in Figure 2a with the tree root being the hip center. Joint angles are parametrized by unit quaternions. The ski joints are fixed such that they are all in a straight line, rotating with the foot. All x, y, z joint coordinates $\boldsymbol{P}_j$ are computed recursively from parent joint $\boldsymbol{P}_i$, starting from the root node c in the following manner:

$$\boldsymbol{P}_j = \boldsymbol{P}_i + \overset{\curvearrowright}{\prod_{k:c\to j}} R_k \begin{bmatrix} b_x^{i\to j} \\ b_y^{i\to j} \\ b_z^{i\to j} \end{bmatrix} \ell(i,j). \tag{1}$$

Here $c \to j$ is the node path from root joint c to joint i,

$$\overset{\curvearrowright}{\prod_{k:c\to j}} R_k = R_c \cdots R_k \cdots R_j \tag{2}$$

denotes matrix multiplication from the right, where $R_i \in \mathbb{R}^{3\times3}$ is the rotation matrix associated with joint i. $b^{i\to j} \in \mathbb{R}^3$ is a base direction of the bone from joint i to joint j with norm 1, while $\ell(i,j)$ is the respective bone length scalar.

An advantage of using joint angles is that it introduces a strong prior on the human pose, which limits it to having a fixed scale. If no restrictions are imposed on the angles, impossible poses can still occur.

4.3.1 Known camera parameters

We assume that the parameters for all frames f and all cameras c are known. Specifically, this means we know the intrinsic matrix $\mathcal{K}^{f,c} \in \mathbb{R}^{3\times3}$, the matrix describing world to camera rotation $\mathcal{R}^{f,c} \in \mathbb{R}^{3\times3}$ and camera location $\boldsymbol{t}^{f,c} \in \mathbb{R}^3$. Using the extrinsics $[\mathcal{R}^{f,c} \mid \boldsymbol{t}^{f,c}]$, the transformation of a world coordinate point $\boldsymbol{P}_j^{f,w}$ to camera c's coordinate frame is given by

$$\boldsymbol{P}_j^{f,c} = \mathcal{R}^{f,c}\boldsymbol{P}_j^{f,w} + \boldsymbol{t}^{f,c}. \tag{3}$$

The projection $\hat{\boldsymbol{p}}_j^{f,c} \in \mathbb{R}^3$ (in homogeneous coordinates) of point $\boldsymbol{P}_j^{f,c}$ onto camera c's image plane is then given by

$$\hat{\boldsymbol{p}}_j^{f,c} = \mathcal{K}^{f,c}\boldsymbol{P}_j^{f,w}. \tag{4}$$

The homogeneous point $\hat{\boldsymbol{p}}_j^{f,c}$ can then be transformed to the Euclidean point $\tilde{\boldsymbol{p}}_j^{f,c} \in \mathbb{R}^2$ by dividing by the last coordinate. Finally, denote the complete projection from world coordinates to an image plane as

$$\pi_c(\boldsymbol{P}_j^{f,w}) = \tilde{\boldsymbol{p}}_j^{f,c}. \tag{5}$$

3D reconstruction is done using a bundle adjustment approach, where we optimize an energy function

$$\arg\min_{\boldsymbol{P}} E(\boldsymbol{P}, \mathcal{K}, \mathcal{R}, \boldsymbol{t}) \tag{6}$$

including a reprojection error, as well as priors on the human body. When using the general pose parametrization, this function is defined as

$$\begin{aligned} E(\boldsymbol{P}, \mathcal{K}, \mathcal{R}, \boldsymbol{t}) =& \lambda_{reproj}E_{reproj} \\ &+ \lambda_{smooth}E_{smooth} \\ &+ \lambda_{limbs}E_{limbs} \end{aligned} \tag{7}$$

and when using joint angles as

$$\begin{aligned} E(\boldsymbol{P}, \mathcal{K}, \mathcal{R}, \boldsymbol{t}) =& \lambda_{reproj}E_{reproj} \\ &+ E_{smooth}. \end{aligned} \tag{8}$$

Reprojection term The 3D joint location estimations are iteratively updated by gradient descent such that when projected to each camera plane, they are as close as possible to the 2D joint locations. If we had perfectly consistent 2D localizations, a simple bundle adjustment process with decent initializations would yield very good results. As this assumptions is not applicable because our 2D detections contain per-joint pixel errors, we have to relax the reprojection errors using a similar method as proposed by Takahashi *et al.* [17]. The reprojection energy term is defined as

$$E_{reproj}(\boldsymbol{P}, \mathcal{K}, \mathcal{R}, \boldsymbol{t}) = \frac{1}{N_F N_C N_J} \sum_{f=1}^{N_F} \sum_{c=1}^{N_C} \sum_{j=1}^{N_J} g(\pi_c(\boldsymbol{P}_j^{f,w}), \boldsymbol{p}_j^{f,c}) \tag{9}$$

The function

$$g(x,y) = (n(0) - n(e_{reproj}(x,y)))e_{reproj}(x,y) \tag{10}$$

relaxes the scaled reprojection errors

$$e_{reproj}(x,y) = ||(x-y)H(y)||_2 \tag{11}$$

where $n(x)$ denotes the normal distribution's probability density function $N(0,\sigma^2)$ and $H(y)$ the heatmap probability value at point y. Using this relaxation with $\sigma^2 = 100$, outlier points up to a certain point do not penalize the energy function too much.

Smoothness term Since we are dealing with human motion from videos, we would like it to be smooth in time. To this end, we would like to minimize the differences in estimated 3D joint velocity between frames.

$$E_{smooth}(\boldsymbol{P}) = \frac{1}{N_F N_J} \sum_{j=1}^{N_J} \sum_{f=1}^{N_F} (\Delta^2[\boldsymbol{P}_j^w](f))^2, \qquad (12)$$

where

$$\Delta^n[\boldsymbol{P}_j^w](f) = \sum_{k=0}^{n} \binom{n}{k} (-1)^{n-k} \boldsymbol{P}_j^w(f+k) \qquad (13)$$

computes the n-th order finite differences over the time dimension of $\boldsymbol{P}_j^w \in \mathbb{R}^{N_F \times 3}$, for any frame $f \in \{1,...,N_F\}$.

Human prior term If we are not using the joint angle representation, we would still like all limbs to consistently have the same lengths over time. To this end, we minimize the difference between the estimated and the known limb lengths,

$$E_{limbs}(\boldsymbol{P}) = \frac{1}{N_F} \sum_{f=1}^{N_F} \sum_{(i,j) \in Limbs} (\|\boldsymbol{P}_i^{f,w} - \boldsymbol{P}_j^{f,w}\|_2 - \ell(i,j))^2. \quad (14)$$

Optimization and parameters When optimizing for absolute 3D positions, all points were initialized in the center between all cameras, with an additional random spread of $\mathscr{U}(-10,10)$ meters. Using the joint angle representation, base poses were initialized in the center with an additional random spread of $\mathscr{U}(-10,10)$ meters, with angles initialized normally as $\mathscr{N}(0,0.1)$. We used the quasi-Newton optimization algorithm *Limited-memory Broyden–Fletcher–Goldfarb–Shanno* (L-BFGS) with learning rate 0.05, running it for 100 epochs, with at most 20 iterations per optimization step. For both the general pose parametrization and the joint angles, the energy terms were scaled by $\lambda_{reproj} = 80$, $\lambda_{smooth} = 1$ and $\lambda_{limbs} = 1$.

4.3.2 Unknown camera rotation

In the same way we optimized the 3D pose positions, it is also possible to freely optimize other parameters such as the camera's rotation matrices. The objective then becomes

$$\arg\min_{\boldsymbol{P},\mathcal{R}} E(\boldsymbol{P},\mathcal{K},\mathcal{R},\boldsymbol{t}). \qquad (15)$$

Like the joint angles, the camera rotations can be parametrized by quaternions and iteratively updated to the correct angles by gradient descent. A problem with this approach is however initialization. If the cameras face in randomly initialized directions, it is unlikely that the optimization objective can converge to a desirable solution. We instead propose an approach where in every gradient descent iteration only the 3D poses are optimized, while the camera are adjusted to always point to the center of estimated poses.

More specifically, in the beginning we initialize all 3D pose positions around the center of all cameras with a random spread of $\mathscr{U}(-1,1)$ meters, with joint angles initialized normally as $\mathscr{N}(0,0.1)$. For every camera we then compute the look-at rotation matrix, with the target being the mean location of the 3D pose and the camera's up direction being the global z-axis. A problem with the look-at matrix is, that the person is originally not necessarily in the middle of the image. To solve this, we first compute the horizontal and vertical relative position of the skier in the 2D image. From the camera intrinsics, we know the Field of View (FoV) and can then pan and tilt the rotation matrix in the opposite direction of the calculated horizontal and vertical FoV shift.

Optimizing with this method for 50 epochs, we get a very rough estimate for the real 3D pose positions and camera rotation matrices, which serves as an initialization for joint optimization of all parameters. We optimize again for 50 epochs, but now also through minimizing the energy/objective function adjust the camera angles. For optimization with free camera rotations, the energy function was weighted with $\lambda_{reproj} = 500$, $\lambda_{smooth} = 1$ and $\lambda_{limbs} = 1$.

5 Results

We report the performance of our algorithms on unseen test samples of both data sets. As metrics, we use the Percentage of Correct Key points (PCK), Mean Per Joint Projection Errors (MPJPE), Center of Mass (CoM) and velocity errors. From the 3D skeleton, coaches could also extract useful biomechanical variables.

5.1 2D pose estimation

We trained OpenPose using four different dataset configurations. First we only trained it on the newly created alpine skiing dataset, which we then augmented with calibration poles. After that we initialized the network using pretrained weights from the MPII Human Pose dataset, and finally we added to that one Kühtai sequence from four camera angles.

In Table 1 we show the Percentage of Correct Key points (PCK), as well as the Mean Per Joint Position Errors (MPJPE) for both datasets, considering both all skier joints and also just the human skeleton. The human pose is considered to be the collection of the 14

Joints used	PCK	MPJPE ± std
Alpine all	92.69	0.0195 ± 0.0638
Alpine body	94.88	0.0132 ± 0.0416
Kühtai all	51.37	0.1064 ± 0.1297
Kühtai body	58.26	0.0835 ± 0.1092

(a) Standard alpine dataset.

Joints used	PCK	MPJPE ± std
Alpine all	95.77	0.0807 ± 0.0488
Alpine body	97.91	0.0625 ± 0.0346
Kühtai all	65.51	0.0137 ± 0.1236
Kühtai body	73.01	0.0092 ± 0.1074

(b) Augmented alpine dataset.

Joints used	PCK	MPJPE ± std
Alpine all	96.76	0.0119 ± 0.1268
Alpine body	98.36	0.0081 ± 0.1077
Kühtai all	70.10	0.0755 ± 0.0429
Kühtai body	76.83	0.0577 ± 0.0268

(c) Augmented alpine dataset with weights initializations trained on MPII dataset.

Joints used	PCK	MPJPE ± std
Alpine all	96.51	0.0119 ± 0.0431
Alpine body	97.81	0.0087 ± 0.0296
Kühtai all	78.11	0.0627 ± 0.1275
Kühtai body	83.12	0.0507 ± 0.1133

(d) Previous configurations, but including one scene with four cameras from Kühtai dataset.

Table 1: 2D pose estimation results with four different dataset configurations used for training.

Metric	Direct 3D	Joint angles
MPJPE [m]	0.096 ± 0.125	0.102 ± 0.188
Centered MPJPE [m]	0.088 ± 0.126	0.110 ± 0.191
Normalized MPJPE [m]	0.0918 ± 0.1225	0.098 ± 0.186
CoM Error [m]	0.05 ± 0.02	0.05 ± 0.02
Speed MAE [m/s]	0.41 ± 0.97	0.43 ± 0.99

(a) With known camera parameters.

Metric	Direct 3D	Joint angles
MPJPE [m]	9.406 ± 6.622	5.523 ± 5.041
Centered MPJPE [m]	0.841 ± 3.295	0.252 ± 0.340
Normalized MPJPE [m]	5.382 ± 4.218	3.912 ± 4.371
CoM Error [m]	11.75 ± 6.38	6.75 ± 5.45
Speed MAE [m/s]	99.34 ± 72.74	32.81 ± 42.61

(b) With unknown camera rotations.

Table 2: Best reconstructed 3D pose metric means and standard deviations when either directly inferring 3D locations or when using joint angles.

joints [0,1,2,3,4,6,7,8,10,11,12,13,14,15] shown in Figure 2a. For the MPJPE, image coordinates between 0 and 1 were used for the joint locations. In the case of the new dataset we have information about joint visibility. Invisible joints were not counted in the PCK results. The data augmentation and taking one Kühtai sequence for training had the most impact on the accuracy.

5.2 3D pose estimation

The 3D poses were evaluated with the same four dataset configurations.

5.2.1 Known camera parameters

In Figure 5 we showcase the performance across datasets, number of cameras and parametrization methods. The method taking all 6 cameras with the OpenPose weights that were trained on both datasets consistently performed the best for both joint angle representation and directly optimizing for 3D coordinates. It's performance is highlighted in Table 2a. Surprisingly, the joint angle representation was in most cases outperformed by the optimization of 3D coordinates. To compute global metrics, such as the CoM and speed, taking 3 instead of 2 cameras gives us the largest performance increase, while using more results in diminishing returns. For the centered MPJPE, 3 cameras also seem to be the minimum.

Metric	Best calibrated	Best uncalibrated	A [10]	B [10]	C [15]
Global MPJPE [m]	0.096 ± 0.125	5.523 ± 5.041	n/a	n/a	n/a
Centered MPJPE [m]	0.088 ± 0.126	0.252 ± 0.340	n/a	n/a	n/a
Normalized MPJPE [m]	0.092 ± 0.123	3.912 ± 4.371	0.122	n/a	0.081
Global CoM Error [m]	0.05 ± 0.02	6.75 ± 5.45	n/a	n/a	n/a
Relative CoM Error [m]	n/a	n/a	0.031	0.034	n/a
Global speed MAE [m/s]	0.41 ± 0.97	32.81 ± 42.61	n/a	n/a	n/a

Table 3: Comparison of best results in calibrated and uncalibrated cases to methods proposed by Ostrek *et al.* [10] (A: *Monocular 3D* and B: *Directly from images*) and Rhodin *et al.* [15] (C: *Semi-supervised*).

Velocity estimation Coaches and athletes desire to get feedback about their increase or potential loss of speed at every moment of the run to enhance their performance. However, optical barriers only provide the average speed within a segment and existing automatic approaches [10] estimate only relative pose metrics, no absolute position nor velocity. Our method allows to estimate the athletes instantaneous velocity as the change in CoM position between two frames. It has a very low mean absolute error of 0.41 m/s and standard deviation of 0.97 m/s.

5.2.2 Unknown camera rotation

The performance of estimating camera rotations using all 6 cameras and the OpenPose weights that were trained on both datasets is shown in Table 2b. The proposed method certainly moved the 3D pose in the right direction but is still far from optimal. It seems to find a relatively reasonable pose for the athlete, but has trouble finding the exact right position in global 3D space. In this case, using joint angles performs much better than when directly optimizing 3D joint locations.

5.2.3 Comparison to existing methods

In Table 3 we compare our best results from both calibrated (using direct 3D joint optimization) and uncalibrated (using joint angles) cases with the methods proposed by Ostrek *et al.* [10] and Rhodin *et al.* [15]. The first method by Ostrek *et al.* estimates monocular 3D poses and computes biomechanical variables from the 3D joint locations, while their second method directly computes all variables from images. Rhodin *et al.* estimate a monocular 3D pose trained using a semi-supervised method. Our best method using calibrated cameras yields on average a very comparable performance to previous work. In addition, we are also able to compute global metrics like the athlete's velocity.

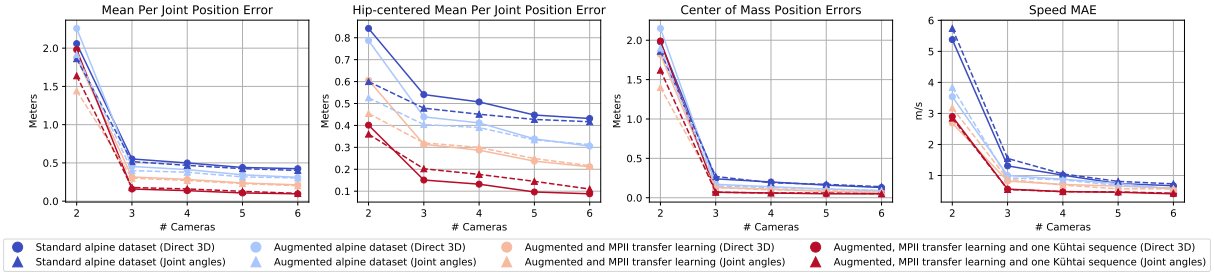


Figure 5: Comparison of all performance metrics for different number of cameras used in bundle adjustment, four differently trained 2D detectors and using joint angles or direct optimization.

6 Conclusion

To sum up our contributions, we created a pipeline taking videos from multiple cameras that is able to reconstruct the global 3D pose of the athlete. To this end, a new alpine skiing dataset was created that can be used for further research in pose estimation. The very specific data augmentation using calibration poles showed measurable performance improvements. This simple to implement augmentation strategy might also translate very well to other datasets that pose similar challenges. We also showed how training OpenPose with different dataset configurations of ever increasing complexity could result in performance improvements in both 2D and 3D pose estimation. From this we can conclude that accurate 2D detections are needed if we want an equally good 3D reconstruction.

When taking fully calibrated cameras, the 3D estimates show a very high CoM and speed accuracy. When estimating camera rotations, the center position is a far way off, so future work could try to incorporate Optical Flow to constrain possible camera movements.

References

[1] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2D Human Pose Estimation: New Benchmark and State of the Art Analysis. In *CVPR*, June 2014.

[2] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields. *CVPR*, 2017.

[3] Jianhui Chen and James J. Little. Sports Camera Calibration via Synthetic Data. *CVPR*, 2018.

[4] Jianhui Chen, Fangrui Zhu, and James J. Little. A Two-point Method for PTZ Camera Calibration in Sports. *WACV*, 2018.

[5] Max deGroot and Ellis Brown. SSD: Single Shot MultiBox Object Detector, in PyTorch. `https://github.com/amdegroot/ssd.pytorch`. [Online; using commit 8dd3865 on Mar 30, 2018].

[6] A Elhayek, Casey Stoll, K I. Kim, and C Theobalt. Outdoor Human Motion Capture by Simultaneous Optimization of Pose and Camera Parameters. *Computer Graphics Forum*, 34, 12 2014.

[7] Benedikt Fasel, Jörg Spörri, Matthias Gilgien, Geo Boffi, Julien Chardonnens, Erich Müller, and Kamiar Aminian. Three-Dimensional Body and Centre of Mass Kinematics in Alpine Ski Racing Using Differential GNSS and Inertial Sensors. *Remote Sensing*, 8, 09 2016.

[8] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6M: Large Scale Datasets and Predictive Methods for 3D Human Sensing in Natural Environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7):1325–1339, jul 2014.

[9] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott E. Reed, Cheng-Yang Fu, and Alexander C. Berg. SSD: Single Shot MultiBox Detector. *ECCV*, 2016.

[10] Mirela Ostrek, Helge Rhodin, Pascal Fua, Erich Müller, and Jörg Spörri. Automating marker-less 3D motion capture with monocular computer vision - a methodological feasibility study on the example of alpine skiing. *Internal technical report*, 2019.

[11] Georgios Pavlakos, Xiaowei Zhou, Konstantinos G. Derpanis, and Kostas Daniilidis. Harvesting Multiple Views for Marker-less 3D Human Pose Annotations. *CVPR*, 2017.

[12] Jens Puwein, Luca Ballan, Remo Ziegler, and Marc Pollefeys. Joint Camera Pose Estimation and 3D Human Pose Estimation in a Multi-camera Setup. In *ACCV*, 2014.

[13] Jens Puwein, Remo Ziegler, Luca Ballan, and Marc Pollefeys. PTZ Camera Network Calibration from Moving People in Sports Broadcasts. *WACV*, pages 25–32, 01 2012.

[14] Joseph Redmon and Ali Farhadi. YOLOv3: An Incremental Improvement. *Technical report*, 2018.

[15] Helge Rhodin, Jörg Spörri, Isinsu Katircioglu, Victor Constantin, Frédéric Meyer, Erich Müller, Mathieu Salzmann, and Pascal Fua. Learning Monocular 3D Human Pose Estimation from Multi-view Images. *CVPR*, 2018.

[16] Jörg Spörri. *Reasearch Dedicated to Sports Injury Prevention - the 'Sequence of Prevention' on the Example of Alpine Ski Racing*. Habilitation with Venia Docendi in "Biomechanics", 2016.

[17] Kosuke Takahashi, Dan Mikami, Mariko Isogawa, and Hideaki Kimata. Human Pose As Calibration Pattern; 3D Human Pose Estimation With Multiple Unsynchronized and Uncalibrated Cameras. In *CVPR Workshops*, June 2018.